Discrete Applied Mathematics 158 (2010) 240–250

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Average-case analysis of incremental topological ordering☆

Deepak Ajwani [a], Tobias Friedrich [b,*]

[a] Center for Massive Data Algorithmics, Aarhus, Denmark

[b] International Computer Science Institute, Berkeley, CA 94704, USA

ARTICLE INFO

Article history:
Received 30 October 2007
Received in revised form 7 February 2009
Accepted 30 July 2009
Available online 12 August 2009

Keywords:
Dynamic graph algorithms
Average-case analysis
Random graphs

ABSTRACT

Many applications like pointer analysis and incremental compilation require maintaining a topological ordering of the nodes of a directed acyclic graph (DAG) under dynamic updates. All known algorithms for this problem are either only analyzed for worst-case insertion sequences or only evaluated experimentally on random DAGs. We present the first average-case analysis of incremental topological ordering algorithms. We prove an expected runtime of $\mathcal{O}(n^2 \text{ polylog}(n))$ under insertion of the edges of a complete DAG in a random order for the algorithms of Alpern et al. (1990) [4], Katriel and Bodlaender (2006) [18], and Pearce and Kelly (2006) [23].

1. Introduction

There has been a growing interest in dynamic graph algorithms over the last two decades due to their applications in a variety of contexts including operating systems, information systems, network management, assembly planning, VLSI design and graphical applications. Typical dynamic graph algorithms maintain a certain property (e.g., connectivity information) of a graph that changes (a new edge inserted or an existing edge deleted) dynamically over time. An algorithm or a problem is called *fully dynamic* if both edge insertions and deletions are allowed, and it is called *partially dynamic* if only one (either only insertion or only deletion) is allowed. If only insertions are allowed, the partially dynamic algorithm is called incremental; if only deletions are allowed, it is called decremental. While a number of fully dynamic algorithms have been obtained for various properties on undirected graphs (see [13] and references therein), the design and analysis of fully dynamic algorithms for directed graphs have turned out to be much harder (e.g., [28,29,27,16]). Much of the research on directed graphs is therefore concentrated on the design of partially dynamic algorithms instead (e.g., [10,5,18]). In this paper, we focus on the analysis of algorithms for maintaining a topological ordering of directed graphs in an incremental setting.

For a directed graph $G = (V, E)$ (with $n := |V|$ and $m := |E|$), a topological order $T: V \to [1 \dots n]$ is a linear ordering of its nodes such that for all directed paths from $x \in V$ to $y \in V$ ($x \neq y$), it holds that $T(x) < T(y)$. A directed graph has a topological ordering if and only if it is acyclic. There are well-known algorithms for computing the topological ordering of a directed acyclic graph (DAG) in $\mathcal{O}(m + n)$ time in an offline setting (see e.g. [11]). In a fully dynamic setting, each time an edge is added or deleted from the DAG, we are required to update the bijective mapping T. In the online/incremental variant of this problem, the edges of the DAG are not known in advance but are inserted one at a time (no deletions allowed). As the topological order remains valid when removing edges, most algorithms for online topological ordering can also handle the fully dynamic setting. However, there are no good bounds known for the topological ordering algorithms in the fully dynamic case. Most algorithms are only analyzed in the incremental setting.

☆ A conference version appeared in the 18th International Symposium on Algorithms and Computation (Ajwani and Friedrich, 2007 [1]). The first author was partially supported by MADALGO — Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation. The second author was partially supported by a postdoctoral fellowship from the German Academic Exchange Service (DAAD).

* Corresponding author. Tel.: +1 510 666 2956.
E-mail address: tobias.friedrich@mpi-inf.mpg.de (T. Friedrich).

0166-218X/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2009.07.006

Given an arbitrary sequence of edges, the online cycle detection problem is to discover the first edge which introduces a cycle. Till now, the best known algorithm for this problem involves maintaining an incremental topological order and returning the edge after which no valid topological order exists. Hence, results for incremental topological ordering also translate into results for the online cycle detection problem. Incremental topological ordering is required for incremental evaluation of computational circuits [4] and in incremental compilation [20,22] where a dependency graph between modules is maintained to reduce the amount of recompilation performed when an update occurs. An application for online cycle detection is pointer analysis [25].

For inserting m edges, the naïve way of computing an incremental topological order each time from scratch with the offline algorithm takes $\mathcal{O}(m^2 + mn)$ time. Marchetti-Spaccamela, Nanni, and Rohnert [21] gave an algorithm that can insert m edges in $\mathcal{O}(mn)$ time. Alpern, Hoover, Rosen, Sweeney, and Zadeck [4] (AHRSZ) proposed an algorithm [4] which runs in $\mathcal{O}(|\rangle\hat{K}\langle| \log(|\rangle\hat{K}\langle|))$ time per edge insertion with $|\rangle\hat{K}\langle|$ being a local measure of the insertion complexity. However, there is no analysis of AHRSZ for a sequence of edge insertions. Katriel and Bodlaender (KB) [18] analyzed a variant of the AHRSZ algorithm and obtained an upper bound of $\mathcal{O}(\min\{m^{\frac{3}{2}} \log n, m^{\frac{3}{2}} + n^2 \log n\})$ for inserting an arbitrary sequence of m edges. The algorithm by Pearce and Kelly (PK) [23] empirically outperforms the other algorithms for random edge insertions leading to sparse random DAGs, although its worst-case runtime is inferior to KB. Ajwani, Friedrich, and Meyer (AFM) [3] proposed a new algorithm with runtime $\mathcal{O}(n^{2.75})$, which asymptotically outperforms KB on dense DAGs.

As noted above, the empirical performance on random edge insertion sequences (REIS) for the above algorithms are quite different from their worst-cases. While PK performs empirically better for REIS, KB and AFM are the best known algorithms for worst-case sequences. This leads us to the theoretical study of incremental topological ordering algorithms on REIS. A nice property of such an average-case analysis is that (in contrast to worst-case bounds) the average of experimental results on REIS converge towards the real average after sufficiently many iterations. This can give a good indication of the tightness of the proven theoretical bounds.

Our contributions are as follows:

- We show an expected runtime of $\mathcal{O}(n^2 \log^2 n)$ for inserting all edges of a complete DAG in a random order with PK (cf. Section 4).
- For AHRSZ and KB, we show an expected runtime of $\mathcal{O}(n^2 \log^3 n)$ for complete random edge insertion sequences (cf. Section 5). This is significantly better than the known worst-case bound of $\mathcal{O}(n^3)$ for KB to insert $\Omega(n^2)$ edges.
- Additionally, we show that for such edge insertion sequences, the expected number of edges which force any algorithm to change the topological order ("invalidating edges") is $\mathcal{O}(n^{\frac{3}{2}}\sqrt{\log n})$ (cf. Section 6), which is the first such result.

A preliminary version of this paper appeared in [1]. The remainder of this paper is organized as follows. The Section 2 describes briefly the three algorithms AHRSZ, KB, and PK. In Section 3 we specify the random graph models used in our analysis. Sections 4 and 6 prove our upper bounds for the runtime of the three algorithms and the number of invalidating edges. Section 7 presents an empirical study, which provides deeper insight on the average-case behavior of AHRSZ and PK.

2. Algorithms

This section first introduces some notations and then describes the three algorithms AHRSZ, KB, and PK. We keep the current topological order as a bijective function $T: V \to [1 \ldots n]$. In this and the subsequent sections, we will use the following notations: $d(u, v)$ denotes $|T(u) - T(v)|$, $u < v$ is a short form of $T(u) < T(v)$, $u \to v$ denotes an edge from u to v, and $u \rightsquigarrow v$ expresses that v is reachable from u. Note that $u \rightsquigarrow u$, but *not* $u \to u$. The *degree* of a node is the sum of its in- and out-degree.

Consider the ith edge insertion $u \to v$. We say that an edge insertion is *invalidating* if $u > v$ before the insertion of this edge. Let $R_B^{(i)}$ denote the set of nodes that have a path to u (in the DAG with the first $i-1$ edges) and whose topological order is greater than or equal to that of v. Also, let $R_F^{(i)}$ denotes the set of nodes that are reachable from v and whose topological order is less than or equal to that of u. Formally, $R_B^{(i)} := \{x \in V \mid (v \leq x) \wedge (x \rightsquigarrow u)\}$, $R_F^{(i)} := \{y \in V \mid (y \leq u) \wedge (v \rightsquigarrow y)\}$ and $\delta^{(i)} = R_F^{(i)} \cup R_B^{(i)}$. Furthermore, let $|\delta^{(i)}|$ denote the number of nodes in $\delta^{(i)}$ and let $\|\delta^{(i)}\|$ denote the number of edges incident to nodes of $\delta^{(i)}$. Note that $\delta^{(i)}$ as defined above is different from the adaptive parameter δ of the bounded incremental computation model. If an edge is non-invalidating, then $|R_B^{(i)}| = |R_F^{(i)}| = |\delta^{(i)}| = 0$. Note that for an invalidating edge, $R_F^{(i)} \cap R_B^{(i)} = \emptyset$ as otherwise the algorithms will just report a cycle and terminate.

We now describe the insertion of the ith edge $u \to v$ for all the three algorithms. Assume for the remainder of this section that $u \to v$ is an invalidating edge, as otherwise none of the algorithms does anything for that edge. We define an algorithm to be *local* if it only changes the ordering of nodes x with $v \leq x \leq u$ to compute the new topological order T' of $G \cup \{(u, v)\}$. All three algorithms are local and they work in two phases — a "discovery phase" and a "relabeling phase".

In the discovery phase of **PK**, the set $\delta^{(i)}$ is identified using a forward depth-first search from v (giving a set $R_F^{(i)}$) and a backward depth-first search from u (giving a set $R_B^{(i)}$). The relabeling phase is also very simple. It sorts both sets $R_F^{(i)}$ and $R_B^{(i)}$ separately in increasing topological order and then allocates new priorities according to the relative position in the sequence

$R_B^{(i)}$ followed by $R_F^{(i)}$. It does not alter the priority of any node not in $\delta^{(i)}$, thereby greatly simplifying the relabeling phase. The runtime of PK for the ith edge insertion is $\Theta(\|\delta^{(i)}\| + |\delta^{(i)}| \log |\delta^{(i)}|)$.

Alpern et al. [4] used the bounded incremental computation model [27] and introduced the measure $|\rangle\hat{K}\langle|$. For an invalidated topological order T, the set $K \subseteq V$ is a *cover* if for all $x, y \in V$: $((x \rightsquigarrow y) \wedge (y < x) \Rightarrow (x \in K) \vee (y \in K))$. This states that for any connected x and y which are incorrectly ordered, a cover K must include x or y or both. $|K|$ and $\|K\|$ denote the number of nodes and edges touching nodes in K, respectively. We define $|\rangle K\langle| := |K| + \|K\|$ and a cover $\hat{K}$ to be *minimal* if $|\rangle\hat{K}\langle| \leq |\rangle K\langle|$ for any other cover K. Thus, $|\rangle\hat{K}\langle|$ captures the minimal amount of work required to calculate the new topological order T' of $G \cup \{(u, v)\}$ assuming that the algorithm is local and that the adjacent edges must be traversed.

AHRSZs *discovery phase* marks the nodes of a cover K by marking some of the unmarked nodes $x, y \in \delta^{(i)}$ with $x \rightsquigarrow y$ and $y < x$. This is done recursively by moving two frontiers starting from v and u towards each other. Here, the crucial decision is which frontier to move next. AHRSZ tries to minimize $\|K\|$ by balancing the number of edges seen on both sides of the frontier. The recursion stops when forward and backward frontier meet. Note that we do not necessarily visit all nodes in $R_F^{(i)}$ ($R_B^{(i)}$) while extending the forward frontier (backward frontier). It can be proven [4] that the marked nodes indeed form a cover K and that $|\rangle K\langle| \leq 3\,|\rangle\hat{K}\langle|$.

The *relabeling phase* employs the dynamic priority space data structure due to Dietz and Sleator [12]. This permits new priorities to be created between existing ones in $\mathcal{O}(1)$ amortized time. This is done in two passes over the nodes in K. During the first pass, it visits the nodes of K in reverse topological order and computes a strict upper bound on the new priorities to be assigned to each node. In the second phase, it visits the nodes in K in topological order and computes a strict lower bound on the new priorities. Both together allow to assign new priorities to each node in K. Thereafter they minimize the number of different labels used to speed up the operations on the priority space data structure in practice. It can be proven that the discovery phase with $|\rangle\hat{K}\langle|$ priority queue operations dominates the time complexity, giving an overall bound of $\mathcal{O}(|\rangle\hat{K}\langle| \log |\rangle\hat{K}\langle|)$.

KB is a slight modification of AHRSZ. In the discovery phase AHRSZ counts the total number of edges incident on a node. KB counts instead only the in-degree of the backward frontier nodes and only the out-degree of the forward frontier nodes. In addition, KB also simplified the relabeling phase. The nodes visited during the extension of the forward (backward) frontier are deleted from the dynamic priority space data structure and are reinserted, in the same relative order among themselves, after (before) all nodes in $R_B^{(i)}$ ($R_F^{(i)}$) not visited during the backward (forward) frontier extension. The algorithm thus computes a cover $K \subseteq \delta^{(i)}$ and its complexity per edge insertion is $\mathcal{O}(|\rangle K\langle| \log |\rangle K\langle|)$. The worst-case running time of KB for a sequence of m edge insertions is $\mathcal{O}(\min\{m^{\frac{3}{2}} \log n, m^{\frac{3}{2}} + n^2 \log n\})$.

3. Random graph model

Erdös and Rényi [14,15] introduced and popularized random graphs. They defined two closely related models: $G(n, p)$ and $G(n, M)$. The $G(n, p)$ model ($0 < p < 1$) consists of a graph with n nodes in which each edge is chosen independently with probability p. On the other hand, the $G(n, M)$ model assigns equal probability to all graphs with n nodes and exactly M edges. Each such graph occurs with a probability of $1/\binom{N}{M}$, where $N := \binom{n}{2}$.

For our study of incremental topological ordering algorithms, we use the random DAG model of Barak and Erdös [6]. They obtain a random DAG by directing the edges of an undirected random graph from lower to higher indexed vertices. Depending on the underlying random graph model, this defines the $DAG(n, p)$ and $DAG(n, M)$ model. We will mainly work on the $DAG(n, M)$ model since it is better suited to describe incremental addition of edges.

The set of all DAGs with n nodes is denoted by DAG^n. For a random variable X with probability space DAG^n, $\mathbf{E}_M[X]$ and $\mathbf{E}_p[X]$ denotes the expected value of X in the $DAG(n, M)$ and $DAG(n, p)$ model, respectively. For the remainder of this paper, we set $\mathbf{E}[X] := \mathbf{E}_M[X]$ and $q := 1 - p$.

The following theorem shows that in most investigations the models $DAG(n, p)$ and $DAG(n, M)$ are practically interchangeable, provided M is close to pN.

Theorem 1. *Given a random variable $X: DAG^n \to [0, a]$ with $a > 0$ and $X(G) \leq X(H)$ for all $G \subseteq H$ and functions p and M of n with $0 < p < 1$ and $M \in \mathbb{N}$.*

(1) *If* $\lim_{n\to\infty} pqN = \lim_{n\to\infty} \frac{pN-M}{\sqrt{pqN}} = \infty$, *then* $\mathbf{E}_M[X] \leq \mathbf{E}_p[X] + o(1)$.

(2) *If* $\lim_{n\to\infty} pqN = \lim_{n\to\infty} \frac{M-pN}{\sqrt{pqN}} = \infty$, *then* $\mathbf{E}_p[X] \leq \mathbf{E}_M[X] + o(1)$.

The analogous theorem for the undirected graph models $G(n, p)$ and $G(n, M)$ is well known. A closer look at the proof for it given by Bollobás [9] reveals that the probabilistic argument used to show the close connection between $G(n, p)$ and $G(n, M)$ can be applied in the same manner for the two random DAG models $DAG(n, p)$ and $DAG(n, M)$.

We define a random edge sequence to be a uniform random permutation of the edges of a complete DAG, i.e., all permutations of $\binom{n}{2}$ edges are equally likely. If the edges appear to the incremental algorithm in the order in which they appear in the random edge sequence, we call it a random edge insertion sequence (REIS). Note that a DAG obtained after

inserting M edges of a REIS will have the same probability distribution as $DAG(n, M)$. To simplify the proofs, we first show our results in the $DAG(n, p)$ model and then transfer them to the $DAG(n, M)$ model by Theorem 1.

4. Analysis of PK

When inserting the ith edge $u \to v$, PK only regards nodes in $\delta^{(i)} := \{x \in V \mid (v \leq x \leq u) \wedge ((v \rightsquigarrow x) \vee (x \rightsquigarrow u))\}$ with "$\leq$" defined according to the current topological order. As discussed in Section 2, PK performs $\mathcal{O}(\|\delta^{(i)}\| + |\delta^{(i)}| \log |\delta^{(i)}|)$ operations for inserting the ith edge. The intuition behind the proofs in this section is that in the early phase of edge insertions (the first $O(n \log n)$ edges), the graph is sparse and so only a few edges are traversed during the DFS traversals. As the graph grows, fewer and fewer nodes are visited in DFS traversals ($|\delta^{(i)}|$ is small) and so the total number of edges traversed in DFS traversals (bounded above by $\|\delta^{(i)}\|$) is still small.

Since PK performs $\mathcal{O}(\|\delta^{(i)}\| + |\delta^{(i)}| \log |\delta^{(i)}|)$ operations for inserting the ith edge, the total cost of PK for inserting all the edges is $O(\sum_{i=1}^{N} \|\delta^{(i)}\| + \sum_{i=1}^{N} (|\delta^{(i)}| \log |\delta^{(i)}|)) = O(\sum_{i=1}^{N} \|\delta^{(i)}\| + \log n \cdot (\sum_{i=1}^{N} |\delta^{(i)}|))$. The last equality follows from the fact that $|\delta^{(i)}| \leq n$ (as $R_F^{(i)} \cap R_B^{(i)} = \emptyset$ for all invalidating edges). Theorems 4 and 10 of this section show for a random edge insertion sequence (REIS) of N edges that $\sum_{i=1}^{N} |\delta^{(i)}| = \mathcal{O}(n^2)$ and $\mathbf{E}\left[\sum_{i=1}^{N} \|\delta^{(i)}\|\right] = \mathcal{O}(n^2 \log^2 n)$. This proves the following theorem.

Theorem 2. *For a random edge insertion sequence (REIS) leading to a complete DAG, the expected runtime of PK is $\mathcal{O}(n^2 \log^2 n)$.*

A comparable pair (of nodes) are two distinct nodes x and y such that either $x \rightsquigarrow y$ or $y \rightsquigarrow x$. We define a potential function Φ_i similar to Katriel and Bodlaender [18]. Let Φ_i be the number of comparable pairs after the insertion of i edges. Clearly,

$$\begin{aligned} &\Delta\Phi_i := \Phi_i - \Phi_{i-1} \geq 0 \quad \text{for all } 1 \leq i \leq M, \\ &\Phi_0 = 0, \quad \text{and} \quad \Phi_M \leq n(n-1)/2. \end{aligned} \tag{1}$$

Theorem 3. *For all edge sequences,* (i) $|\delta^{(i)}| \leq \Delta\Phi_i + 1$ *and* (ii) $|\delta^{(i)}| \leq 2\Delta\Phi_i$.

Proof. Consider the ith edge (u, v). If $u < v$, the theorem is trivial since $|\delta^{(i)}| = 0$. Otherwise, each vertex of $R_F^{(i)}$ and $R_B^{(i)}$ (as defined in Section 2) gets newly ordered with respect to u and v, respectively. The set $\bigcup_{x \in R_B^{(i)}} (x, v) \cap \bigcup_{x \in R_F^{(i)}} (u, x) = \{(u, v)\}$. This means that overall at least $|R_F^{(i)}| + |R_B^{(i)}| - 1$ node pairs get newly ordered:

$$\Delta\Phi_i \geq |R_F^{(i)}| + |R_B^{(i)}| - 1 = |\delta^{(i)}| - 1.$$

Also, since in this case $\Delta\Phi_i \geq 1$, $|\delta^{(i)}| \leq 2\Delta\Phi_i$. □

Theorem 4. *For all edge sequences,* $\sum_{i=1}^{N} |\delta^{(i)}| \leq n(n-1) = \mathcal{O}(n^2)$.

Proof. By Theorem 3(i), we get $\sum_{i=1}^{N} |\delta^{(i)}| \leq \sum_{i=1}^{N} (\Delta\Phi_i + 1) = \Phi_N + N \leq n(n-1)/2 + n(n-1)/2 = n(n-1)$. □

The remainder of this section provides the necessary tools step by step to finally prove the desired bound on $\sum_{i=1}^{N} \|\delta^{(i)}\|$ in Theorem 10. One can also interpret Φ_i as a random variable in $DAG(n, M)$ with $M = i$. The corresponding function Ψ for $DAG(n, p)$ is defined as the total number of comparable node pairs in $DAG(n, p)$. Pittel and Tungol [26] showed the following theorem.

Theorem 5. *For $p := c \log(n)/n$ and $c > 1$,* $\mathbf{E}_p[\Psi] = (1 + o(1)) \frac{n^2}{2} \left(1 - \frac{1}{c}\right)^2$.

Using Theorem 1, this result can be transformed to Φ as defined above for $DAG(n, M)$ and gives the following bounds for $\mathbf{E}_M[\Phi_k]$.

Theorem 6. *For $n \log n < k \leq N - 2n \log n$,*

$$\mathbf{E}_M[\Phi_k] = (1 + o(1)) \frac{n^2}{2} \left(1 - \frac{(n-1) \log n}{2(k + n \log n)}\right)^2.$$

For $N - 2n \log n < k \leq N - 2 \log n$,

$$\mathbf{E}_M[\Phi_k] = (1 + o(1)) \frac{n^2}{2} \left(1 - \frac{(n-1) \log n}{2(k + \sqrt{\log n\,(N-k)})}\right)^2.$$

Proof. The function $\Psi: DAG^n \to [0, N]$ satisfies $\Psi(G) \leq \Psi(H)$ wherever $G \subseteq H$. The later inequality is true as the nodes already ordered in G will still remain ordered in H. For $n \log n < k \leq N - 2n \log n$, consider $p := \frac{k+n \log n}{N}$. Then

$$\lim_{n\to\infty} pqN \geq \lim_{n\to\infty} \frac{\log n}{n} \frac{\log n}{n} N \geq \lim_{n\to\infty} \frac{(n-1)\log^2 n}{2n} = \infty$$

and

$$\begin{aligned}\lim_{n\to\infty} \frac{pN-k}{\sqrt{pqN}} &\geq \lim_{n\to\infty} \frac{pN-k}{\sqrt{N}} \geq \lim_{n\to\infty} \frac{n\log n}{\sqrt{N}} \\ &\geq \lim_{n\to\infty} \frac{n\log n}{n} \geq \lim_{n\to\infty} \log n = \infty.\end{aligned}$$

Since all the conditions of Theorem 1 are satisfied for these values of k and p, $\mathbf{E}_M[\Psi] = O(\mathbf{E}_p[\Psi])$. In particular,

$$\mathbf{E}_M[\Phi_k] = \mathbf{E}_{p=(k+n\log n)/N}[\Psi] + o(1) = (1+o(1))\frac{n^2}{2}\left(1 - \frac{(n-1)\log n}{2(k+n\log n)}\right)^2.$$

For $N - 2n \log n < k \leq N - 2 \log n$, we choose $p := \frac{k+\sqrt{\log n\,(N-k)}}{N}$. Clearly,

$$p \geq \frac{N - 2n\log n + \sqrt{\log n(N-(N-2\log n))}}{N} \geq \frac{N - 2n\log n + \sqrt{2}\log n}{N}.$$

Using this, we get

$$\lim_{n\to\infty} pqN \geq \lim_{n\to\infty} \frac{(N - 2n\log n + \sqrt{2}\log n)}{N} \frac{(N - k - \sqrt{\log n\,(N-k)})\,N}{N}.$$

Observe that $f(k) := N - k - \sqrt{\log n\,(N-k)}$ has its minimum at $k_0 = N - \log(n)/4$ since $f'(k_0) = 0$ and $f''(k_0) = 2/\log n > 0$. Hence, we conclude that $f(k)$ is monotonically decreasing in our interval $(N - 2n \log n, N - 2 \log n)$ and attains its minimum at $N - 2 \log n$. Therefore, $N - k - \sqrt{\log n\,(N-k)} \geq 2\log n - \sqrt{2}\log n \to \infty$, which in turn proves $\lim_{n\to\infty} pqN = \infty$ and

$$\lim_{n\to\infty} \frac{pN-k}{\sqrt{pqN}} \geq \lim_{n\to\infty} \frac{\sqrt{\log n\,(N-k)}}{\sqrt{N-k-\sqrt{\log n\,(N-k)}}} \geq \lim_{n\to\infty} \sqrt{\log n} = \infty$$

Together with Theorem 5, this yields

$$\begin{aligned}\mathbf{E}_M[\Phi_k] &= \mathbf{E}_{p=(k+\sqrt{\log n\,(N-k)})/N}[\Psi] + o(1) \\ &= (1+o(1))\,\frac{n^2}{2}\left(1 - \frac{(n-1)\log n}{2(k+\sqrt{\log n\,(N-k)})}\right)^2. \quad \square\end{aligned}$$

The degree sequence of a random graph is a well-studied problem. The following theorem is shown in [9].

Theorem 7. *If $pn/\log n \to \infty$, then almost every graph G in the $G(n, p)$ model satisfies $\Delta(G) = (1 + o(1))\,pn$, where $\Delta(G)$ is the maximum degree of a node in G.*

As noted in Section 3, the undirected graph obtained by ignoring the directions of $DAG(n, p)$ is a $G(n, p)$ graph. Therefore, the above result is also true for the maximum degree (in-degree + out-degree) of a node in $DAG(n, p)$. Using Theorem 1, the above result can be transformed to $DAG(n, M)$, as well.

Theorem 8. *With probability $1 - \mathcal{O}(\frac{1}{n})$, there is no node with degree higher than $21\frac{M}{n}$ for sufficiently large n and $M > n \log n$ in $DAG(n, M)$.*

Proof. We examine the following two functions:

- $f_1(g)$: Number of nodes with degree at least $g(n)$
- $f_2(g) := f_1^2(g)$.

For f_1, f_2 in $G(n, p)$, $g(n) := pn + 2\sqrt{pqn\log n}$, and some constant c, Bollobás [8] showed

$$\begin{aligned}&\mathbf{E}_p[f_1(g)] = \mathcal{O}\left(\frac{1}{n}\right), \\ &\sigma_p^2(f_1(g)) = \mathbf{E}_p[f_2(g)] - \mathbf{E}_p^2[f_1(g)] \leq c \cdot \mathbf{E}_p[f_1(g)].\end{aligned} \tag{2}$$

Consider any random $DAG(n, M)$. It must have been obtained by taking a random graph $G(n, M)$ and ordering the edges. The degree of a node in $DAG(n, M)$ is the same as the degree of the corresponding node in $G(n, M)$.

We break down the analysis depending on M. At first, consider the simpler case of $M > \left(\lfloor \frac{N}{n \log n} \rfloor - 2\right) n \log n$. The degree of any node in an undirected graph cannot be higher than $n-1$. However, as $M > N - 3n \log n$, $21 \cdot \frac{M}{n} \geq \frac{21}{2}(n-1) - 63 \log n$. For sufficiently large n this is greater than $n-1$ and therefore, no node can have degree higher than it.

Next, we consider $M \in (kn \log n, (k+1)\, n \log n]$ for $1 \leq k < l$, where $l := \lfloor \frac{N}{n \log n} \rfloor - 2$, and we prove the theorem for each interval. We choose $p_k := (k+2) \frac{n \log n}{N}$, $q_k := 1 - p_k$, and $g_k(n) := p_k n + 2\sqrt{p_k q_k n \log n}$ and look for the conditions in Theorem 1. Note that $0 < p_k < 1$, $f_1 \colon G^n \to [0, n]$, $f_2 \colon G^n \to [0, n^2]$, and $f_i(G) \leq f_i(H)$ wherever $G \subseteq H$ for $i = 1, 2$. The later inequality holds as the degree of any node in H is greater than or equal to the corresponding degree in G. For $1 \leq k < l$,

$$p_k \geq \frac{3n \log n}{N} \geq \frac{6 \log n}{n-1}$$

and

$$q_k \geq 1 - \left(\left\lfloor \frac{N}{n \log n} \right\rfloor - 1\right) \frac{n \log n}{N} \geq 1 - \left(\frac{N - n \log n}{n \log n}\right) \frac{n \log n}{N} \geq \frac{2 \log n}{n-1}.$$

So for each interval,

$$\lim_{n \to \infty} p_k q_k N \geq \lim_{n \to \infty} \frac{6 \log n}{n-1} \frac{2 \log n}{n-1} N \geq \lim_{n \to \infty} 6 \log^2 n = \infty$$

and by $M_k \leq (k+1)\, n \log n$ and $k + 2 \leq \lfloor \frac{N}{n \log n} \rfloor$,

$$\begin{aligned} \lim_{n \to \infty} \frac{pN - M}{\sqrt{pqN}} &\geq \lim_{n \to \infty} \frac{pN - M}{\sqrt{pN}} \geq \lim_{n \to \infty} \frac{n \log n}{\sqrt{(k+2)\, n \log n}} = \lim_{n \to \infty} \frac{\sqrt{n \log n}}{\sqrt{k+2}} \\ &\geq \lim_{n \to \infty} \frac{n \log n}{\sqrt{N}} \geq \lim_{n \to \infty} \log n = \infty. \end{aligned}$$

In each interval, all the conditions of Theorem 1 are satisfied and therefore, $\mathbf{E}_M[f_i(g_k)] = \mathbf{E}_{p_k}[f_i(g_k)] + o(1)$ for $i = 1, 2$ and $1 \leq k < l$. Using Eq. (2), we get $\mathbf{E}_M[f_1(g_k)] = \mathcal{O}(\mathbf{E}_{p_k}[f_1(g_k)]) = \mathcal{O}\left(\frac{1}{n}\right)$ and

$$\begin{aligned} \sigma^2_M(f_1(g_k)) &= \mathbf{E}_M[f_2(g_k)] - \mathbf{E}^2_M[f_1(g_k)] = \mathcal{O}\left(\mathbf{E}_{p_k}[f_2(g_k)] - \mathbf{E}^2_{p_k}[f_1(g_k)]\right) \\ &= \mathcal{O}(\sigma^2_{p_k}(f_1(g_k))) = \mathcal{O}(\mathbf{E}_{p_k}[f_1(g_k)]) = \mathcal{O}\left(\frac{1}{n}\right). \end{aligned}$$

Therefore, by substituting $X := f_1(g_k)$, $\mu := \mathbf{E}_M[f_1(g_k)] = \mathcal{O}\left(\frac{1}{n}\right)$, $\sigma^2 := \sigma^2_M(f_1(g_k)) = \mathcal{O}\left(\frac{1}{n}\right)$, and $\nu := 1 - \mu$ in Chebyshev's inequality ($\Pr\{|X - \mu| \geq \nu\} \leq \frac{\sigma^2}{\nu^2}$), we get

$$\Pr\{|f_1(g_k) - \mu| \geq 1 - \mu\} \leq \mathcal{O}\left(\frac{1}{n(1-\mu)^2}\right) = \mathcal{O}\left(\frac{1}{n}\right).$$

However, $\Pr\{|f_1(g_k) - \mu| \geq 1 - \mu\} = \Pr\{(f_1(g_k) \geq 1) \text{ or } (f_1(g_k) \leq 2\mu - 1)\}$ and since, $\mu = \mathcal{O}\left(\frac{1}{n}\right)$ and $f_1(g_k)$ is non-negative random variable, $\Pr\{f_1(g_k) \leq 2\mu - 1\} = 0$ for sufficiently large n. Therefore, $\Pr\{f_1(g_k) \geq 1\} = \Pr\{|f_1(g_k) - \mu| \geq 1 - \mu\} = \mathcal{O}\left(\frac{1}{n}\right)$. In other words, with probability $(1 - \mathcal{O}(\frac{1}{n}))$, there is no node with a degree higher than g_k in any interval. However, by $p_k \geq \frac{\log n}{n}$ we get

$$g_k(n) = p_k n + 2\sqrt{p_k q_k n \log n} \leq 3 p_k n \leq 6(k+2) \frac{n \log n}{n-1}.$$

For sufficiently large n, $\frac{n}{n-1} \leq \frac{7}{6}$, and this implies

$$g_k(n) \leq 7(k+2) \log n \leq \frac{7(k+2)}{k} \frac{M}{n} \leq \frac{21M}{n}.$$

Therefore, with probability $1 - \mathcal{O}(\frac{1}{n})$, there is no node with a degree higher than $21\frac{M}{n}$ in $G(n, M)$ and by the argument above, in $DAG(n, M)$. □

As the maximum degree of a node in $DAG(n, i)$ is $\mathcal{O}(i/n)$, we finally just need to show a bound on $\sum_i (i \cdot |\delta^{(i)}|)$ to prove Theorem 10. This is done in the following theorem.

Theorem 9. *For DAG(n, M) and $r := N - 2\log n$,*

$$\mathbf{E}\left[\sum_{i=1}^{r}(i \cdot |\delta^{(i)}|)\right] = \mathcal{O}(n^3 \log^2 n).$$

Proof. Let us decompose the analysis in three steps. First, we show a bound on the first $n \log n$ edges. By definition of $\delta^{(i)}$, $|\delta^{(i)}| \leq n$. Therefore,

$$\sum_{i=1}^{n\log n} i \cdot \mathbf{E}\left[|\delta^{(i)}|\right] \leq \sum_{i=1}^{n\log n} i \cdot n = \mathcal{O}\left(n^3 \log^2 n\right). \tag{3}$$

The second step is to bound $\sum_{i=n\log n}^{t} i \cdot |\delta^{(i)}|$ with $t := N - 2n\log n$. For this, Theorem 3(ii) shows for all k such that $n\log n < k < t$ that

$$\mathbf{E}\left[\sum_{i=k}^{t}|\delta^{(i)}|\right] \leq 2\,\mathbf{E}\left[\sum_{i=k}^{t}\Delta\Phi_i\right] = 2\,\mathbf{E}\left[\Phi_t - \Phi_{k-1}\right] = 2\,\mathbf{E}\left[\Phi_t\right] - 2\mathbf{E}\left[\Phi_{k-1}\right]. \tag{4}$$

The function hidden in the $o(1)$ in Theorem 5 is decreasing in p [26]. Hence, also the $o(1)$ in Theorem 6 is decreasing in k. Plugging this in Eq. (4) yields (with $s := n\log n$ and using $k < t$)

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=k}^{t}|\delta^{(i)}|\right] &\leq (1+o(1))\,n^2\left(\left(1-\frac{(n-1)\log n}{2(t+s)}\right)^2 - \left(1-\frac{(n-1)\log n}{2(k-1+s)}\right)^2\right)\\
&= (1+o(1))\,n^2(n-1)\log n\left(\frac{2}{2(k-1+s)} - \frac{2}{2(t+s)}\right.\\
&\quad \left. + \frac{(n-1)\log n}{4}\left(\frac{1}{(t+s)^2} - \frac{1}{(k-1+s)^2}\right)\right)\\
&\leq (1+o(1))\,n^2(n-1)\log n\left(\frac{1}{k-1+s} - \frac{1}{t+s}\right)\\
&\leq (1+o(1))\,n^2(n-1)\log n\frac{1}{k-1}.
\end{aligned} \tag{5}$$

By linearity of expectation and Eq. (5),

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=s+1}^{t} i\,|\delta^{(i)}|\right] &= \sum_{i=s+1}^{t}\left(i\,\mathbf{E}\left[|\delta^{(i)}|\right]\right) \leq \sum_{j=1}^{\log(\lceil\frac{t}{s}\rceil)}\left(2^j s \sum_{i=2^{(j-1)}s+1}^{2^j s}\mathbf{E}\left[|\delta^{(i)}|\right]\right)\\
&\leq \sum_{j=1}^{\log(\lceil\frac{t}{s}\rceil)}\left(2^j s \sum_{i=2^{(j-1)}s+1}^{t}\mathbf{E}\left[|\delta^{(i)}|\right]\right)\\
&\leq \sum_{j=1}^{\log(\lceil\frac{t}{s}\rceil)}\left(2^j s(1+o(1))\,n^2(n-1)\log n\frac{1}{2^{(j-1)}s}\right)\\
&= \sum_{j=1}^{\log(\lceil\frac{t}{s}\rceil)}\left(2(1+o(1))\,n^2(n-1)\log n\right)\\
&= 2(1+o(1))\,n^2(n-1)\log^2 n = \mathcal{O}(n^3\log^2 n).
\end{aligned}$$

For the last step consider a k such that $t < k < r$. Theorem 3(ii) gives

$$\mathbf{E}\left[\sum_{i=k}^{r}|\delta^{(i)}|\right] \leq 2\,\mathbf{E}\left[\sum_{i=k}^{r}\Delta\Phi_i\right] = 2\,\mathbf{E}\left[\Phi_r - \Phi_{k-1}\right] = 2\,\mathbf{E}\left[\Phi_r\right] - 2\mathbf{E}\left[\Phi_{k-1}\right].$$

Using Theorem 6 and similar arguments as before, this yields (with $s(k) := \sqrt{\log n\,(N-k)}$)

$$\begin{aligned}\mathbf{E}\left[\sum_{i=k}^{r}|\delta^{(i)}|\right] &\leq (1+o(1))\,n^2\left(\left(1-\frac{(n-1)\log n}{2(r+s(r))}\right)^2-\left(1-\frac{(n-1)\log n}{2(k-1+s(k-1))}\right)^2\right)\\ &= (1+o(1))\,n^2(n-1)\log n\left(\frac{2}{2(k-1+s(k-1))}-\frac{2}{2(r+s(r))}\right.\\ &\quad\left.+\frac{(n-1)\log n}{4}\left(\frac{1}{(r+s(r))^2}-\frac{1}{(k-1+s(k-1))^2}\right)\right).\end{aligned}$$

Since $k+s(k)$ is monotonically increasing for $t<k<r$, $\frac{1}{(k+s(k))^2}$ is a monotonically decreasing function in this interval. Therefore, $\frac{1}{(r+s(r))^2}-\frac{1}{(k-1+s(k-1))^2}<0$, which proves the following equation.

$$\begin{aligned}\mathbf{E}\left[\sum_{i=k}^{r}|\delta^{(i)}|\right] &\leq (1+o(1))\,n^2(n-1)\log n\left(\frac{1}{k-1+s(k-1)}-\frac{1}{r+s(r)}\right)\\ &\leq (1+o(1))\,n^2(n-1)\log n\frac{1}{k-1}.\end{aligned}\tag{6}$$

By linearity of expectation and Eq. (6),

$$\begin{aligned}\mathbf{E}\left[\sum_{i=N-2n\log n+1}^{r} i\,|\delta^{(i)}|\right] &= \sum_{i=N-2n\log n+1}^{r}\left(i\,\mathbf{E}\left[|\delta^{(i)}|\right]\right)\\ &\leq (N-2\log n)\sum_{i=N-2n\log n+1}^{r}\mathbf{E}\left[|\delta^{(i)}|\right]\\ &\leq (N-2\log n)\,(1+o(1))\,n^2(n-1)\log n\frac{1}{N-2n\log n-1}\\ &= \mathcal{O}(n^3\log n).\quad\square\end{aligned}$$

Theorem 10. *For DAG(n, M),* $\mathbf{E}\left[\sum_{i=1}^{N}\|\delta^{(i)}\|\right]=\mathcal{O}(n^2\log^2 n)$.

Proof. By definition of $\|\delta^{(i)}\|$, we know $\|\delta^{(i)}\|\leq i$ and hence

$$\sum_{i=1}^{n\log n}\|\delta^{(i)}\|=\mathcal{O}(n^2\log^2 n).$$

Again, let $r := N-2\log n$. Theorem 8 tells us that with probability greater than $\left(1-\frac{c'}{n}\right)$ for some constant c', there is no node with degree $\geq \frac{c\,i}{n}$ (for $c=21$). Since the degree of an arbitrary node in a DAG is bounded by n, we get with Theorems 4 and 9,

$$\begin{aligned}\mathbf{E}\left[\sum_{i=n\log n+1}^{r}\|\delta^{(i)}\|\right] &= \mathcal{O}\left(\mathbf{E}\left[\sum_{i=n\log n+1}^{r}\frac{c\,i\,|\delta^{(i)}|}{n}\right]+\mathbf{E}\left[\sum_{i=n\log n+1}^{r}\frac{n\,c'\,|\delta^{(i)}|}{n}\right]\right)\\ &= \mathcal{O}\left(\frac{1}{n}\,\mathbf{E}\left[\sum_{i=1}^{r}(i\,|\delta^{(i)}|)\right]+n^2\right)\\ &= \mathcal{O}\left(\frac{1}{n}\left(n^3\log^2 n\right)+n^2\right)=\mathcal{O}(n^2\log^2 n).\end{aligned}$$

By again using the fact that the degree of an arbitrary node in a DAG is at most n, we obtain

$$\mathbf{E}\left[\sum_{i=r+1}^{N}\|\delta^{(i)}\|\right]=\mathcal{O}\left(n\cdot\mathbf{E}\left[\sum_{i=r+1}^{N}|\delta^{(i)}|\right]\right)=\mathcal{O}\left(n\cdot\sum_{i=r+1}^{N}n\right)=\mathcal{O}(n^2\log n).$$

Thus,

$$\mathbf{E}\left[\sum_{i=1}^{N}\|\delta^{(i)}\|\right] = \mathbf{E}\left[\sum_{i=1}^{n\log n}\|\delta^{(i)}\|\right] + \mathbf{E}\left[\sum_{i=n\ \log n+1}^{r}\|\delta^{(i)}\|\right] + \mathbf{E}\left[\sum_{i=r+1}^{N}\|\delta^{(i)}\|\right]$$
$$= \mathcal{O}(n^2\log^2 n) + \mathcal{O}(n^2\log^2 n) + \mathcal{O}(n^2\log n) = \mathcal{O}(n^2\ \log^2 n). \quad \square$$

5. Analysis of AHRSZ and KB

Katriel and Bodlaender [18] introduced KB as a variant of AHRSZ for which a worst-case runtime of $\mathcal{O}(\min\{m^{\frac{3}{2}}\log n, m^{\frac{3}{2}} + n^2\log n\})$ can be shown. In this section, we prove an expected runtime of $\mathcal{O}(n^2\log^3 n)$ under random edge insertion sequences, both for AHRSZ and KB.

Recall from Section 2 that for every edge insertion there is a minimal cover $\hat{K}^{(i)}$. The following theorem shows that $\delta^{(i)}$ is also a valid cover in this situation.

Theorem 11. *$\delta^{(i)}$ is a valid cover.*

Proof. Consider the insertion of the ith edge (u, v) and consider a node-pair x, y such that $x \rightsquigarrow y$, but $x > y$. Since before the insertion of this edge, the topological ordering was consistent, $x \rightsquigarrow u \to v \rightsquigarrow y, x < u$ and $v < y$. Together with $x > y$, it implies $x > v$. Now $x \rightsquigarrow u$ and $x \geq v$ imply $x \in \delta^{(i)}$. Thus, for every node-pair (x, y) such that $x \rightsquigarrow y$ and $x > y, x \in \delta^{(i)}$ and hence, $\delta^{(i)}$ is a valid cover. $\square$

Therefore, by definition of $|\rangle\hat{K}^{(i)}\langle|$, $|\rangle\hat{K}^{(i)}\langle| \leq |\rangle\delta^{(i)}\langle| = |\delta^{(i)}| + \|\delta^{(i)}\|$.

$$\mathbf{E}\left[\sum_{i=1}^{m}|\rangle\hat{K}^{(i)}\langle|\right] \leq \sum_{i=1}^{m}|\delta^{(i)}| + \mathbf{E}\left[\sum_{i=1}^{m}\|\delta^{(i)}\|\right] = \mathcal{O}(n^2\log^2 n).$$

The latter equality follows from Theorems 4 and 10. The expected complexity of AHRSZ on REIS is thus $\mathcal{O}\left(\mathbf{E}\left[\sum_{i=1}^{m}|\rangle\hat{K}^{(i)}\langle|\log n\right]\right) = \mathcal{O}(n^2\log^3 n)$.

KB also computes a cover $K \subseteq \delta^{(i)}$ and its complexity per edge insertion is $\mathcal{O}(|\rangle K\langle|\ \log|\rangle K\langle|)$. Therefore, $|\rangle K\langle| \leq |\delta^{(i)}| + \|\delta^{(i)}\|$ and with a similar argument as above, the expected complexity of KB on REIS is $\mathcal{O}(n^2\log^3 n)$.

6. Bounding the number of invalidating edges

An interesting question in all this analysis is how many edges will actually invalidate the topological ordering and force any algorithm to do something about them. Here, we show a non-trivial upper bound on the expected value of the number of invalidating edges on REIS. Consider the following random variable: $\textsc{inval}(i) = 1$ if the ith edge inserted is an invalidating edge; $\textsc{inval}(i) = 0$ otherwise.

Theorem 12. $\mathbf{E}\left[\sum_{i=1}^{m}\textsc{inval}(i)\right] = \mathcal{O}(\min\{m, n^{\frac{3}{2}}\log^{\frac{1}{2}} n\})$.

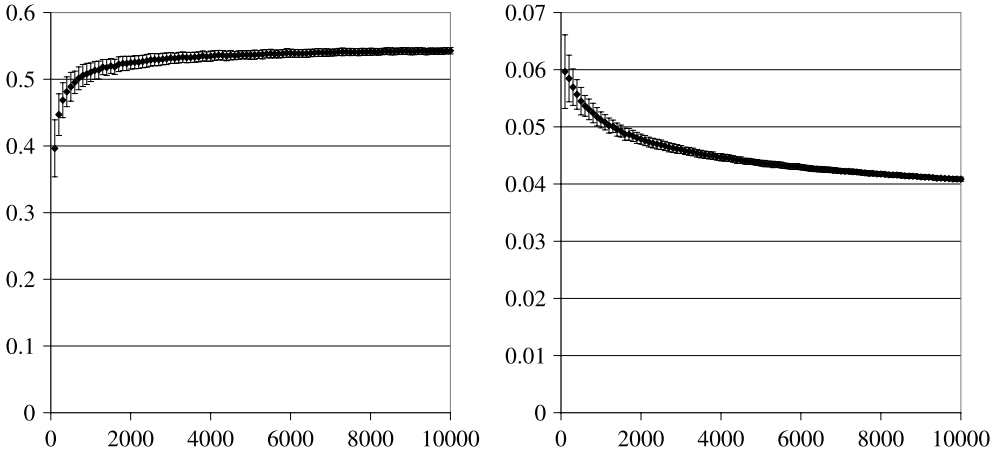
Proof. If the ith edge is invalidating, $|\delta^{(i)}| \geq 2$; otherwise $\textsc{inval}(i) = |\delta^{(i)}| = 0$. In either case, $\textsc{inval}(i) \leq |\delta^{(i)}|/2$. Thus, for $s := n^{\frac{3}{2}}\log^{\frac{1}{2}} n$ and $t := \min\{m, N - 2n\log n\}$,

$$\mathbf{E}\left[\sum_{i=s+1}^{t}\textsc{inval}(i)\right] \leq \mathbf{E}\left[\sum_{i=s+1}^{t}\frac{|\delta^{(i)}|}{2}\right] \leq (1 + o(1))\,\frac{n^2(n-1)\log n}{2s}$$
$$\leq \frac{(1 + o(1))}{2}\,n^{\frac{3}{2}}\log^{\frac{1}{2}} n.$$

The second inequality follows by substituting $k := s + 1$ in Eq. (5). Also, since the number of invalidating edges can be at most equal to the total number of edges, $\sum_{i=1}^{s}\textsc{inval}(i) \leq s$.

$$\mathbf{E}\left[\sum_{i=1}^{m}\textsc{inval}(i)\right] = \mathbf{E}\left[\sum_{i=1}^{s}\textsc{inval}(i)\right] + \mathbf{E}\left[\sum_{i=s+1}^{t}\textsc{inval}(i)\right] + \mathbf{E}\left[\sum_{i=t}^{m}\textsc{inval}(i)\right]$$
$$\leq \mathcal{O}(s) + \mathcal{O}(n^{\frac{3}{2}}\log^{\frac{1}{2}} n) + \mathcal{O}(n\log n) = \mathcal{O}(n^{\frac{3}{2}}\log^{\frac{1}{2}} n).$$

The second bound $\mathbf{E}\left[\sum_{i=1}^{m}\textsc{inval}(i)\right] \leq m$ is obvious by definition of $\textsc{inval}(i)$. $\square$



(a) $C(n)/(n^2 \log n)$. (b) $C(n)/(n^2 \log^2 n)$.

Fig. 1. Experimental results of AHRSZ for the insertion of the edges of a complete DAG in a random order. The horizontal axes describe the number of vertices n. The vertical axes show the measured empirical insertion costs $C(n) := \sum_i |\rangle K\langle| \log |\rangle K\langle|$ relative to (a) $n^2 \log n$ and (b) $n^2 \log^2 n$, respectively. The error bars specify the sample standard deviation.

7. Empirical observations

In addition to the achieved average-case bounds, we also examined AHRSZ and PK experimentally using the implementation of Pearce [23] available from www.mcs.vuw.ac.nz/~djp/dts.html. For varying number of vertices $n = 100, 200, \ldots, 10000$, we generated random edge insertion sequences (REIS) leading to complete DAGs and averaged the performance parameter $C(n)$ over 250 runs. The chosen $C(n)$ upper bounds the respective runtimes.

The performance parameter taken for AHRSZ is $C(n) := \sum_i |\rangle K\langle| \log(|\rangle K\langle|)$. We know $\mathbf{E}[C(n)] = \mathcal{O}(n^2 \log^3 n)$ from Section 5 and know that the overall runtime is $\Omega(n^2)$ since the algorithm has to inspect all the edges being inserted. In our experimental setting, we discovered that $C(n)/(n^2 \log^2 n)$ is apparently a decreasing function and that $C(n)/(n^2 \log n)$ is an increasing function. This empirical evidence suggests that $C(n)$ is possibly between $\Omega(n^2 \log n)$ and $\mathcal{O}(n^2 \log^2 n)$. Fig. 1 shows our experimental results for AHRSZ.

We consider $C(n) := \sum_i (\|\delta^{(i)}\| + |\delta^{(i)}| \log |\delta^{(i)}|)$ as a performance parameter for PK and observe that $C(n)/n^2$ is decreasing while $C(n)/(n^2 \log^{-1} n)$ is increasing. This indicates that $C(n) = o(n^2)$, which implies an actual runtime of $\Theta(n^2)$ for PK on REIS since all $\Omega(n^2)$ edges have to be inspected. Pearce and Kelly [23] showed empirically that PK outperforms AHRSZ on sparse DAGs. Our experiments extend this to dense DAGs.

Complementing Section 6, we also examined empirically the number of invalidating edges for AHRSZ. The same experimental set-up as above suggests a quasilinear growth of $\sum_{i=1}^{m} \textsc{inval}(i)$ between $\Omega(n \log n)$ and $\mathcal{O}(n \log^2 n)$. Note that the observed empirical bound for AHRSZ is significantly lower than the general bound $\mathcal{O}(n^{\frac{3}{2}} \log^{\frac{1}{2}} n)$ of Theorem 12 which holds for all algorithms.

8. Discussion

On random edge insertion sequences (REIS) leading to a complete DAG, we have shown an expected runtime of $\mathcal{O}(n^2 \log^2 n)$ for the incremental topological ordering algorithm PK and $\mathcal{O}(n^2 \log^3 n)$ for the algorithms AHRSZ and KB while the trivial lower bound is $\Omega(n^2)$. Our analysis can be adapted to prove that for $n \log n < f(n) < N - n \log n$, the expected time to insert m edges (in the REIS) after $f(n)$ edges have already been inserted (in the REIS) is $\mathcal{O}(\frac{n^2 m \log n}{f(n)})$. In particular, if $\mathcal{O}(n^2)$ edges have already been inserted in a REIS, inserting the next m edges of the REIS will only require $\mathcal{O}(m \log n)$ time. This result is, however, only relevant when inserting edges in a random dense DAG. In order to prove interesting results for the sparse case as well, one needs a deeper analysis of the relationship between $|\delta^{(i)}|$ and $\Delta\Phi_i$ for sparse random DAGs. This, however, remains an interesting open problem.

Recently, there has been a lot of work on improving the worst-case upper bound for incremental topological ordering. Haeupler et al. [17] gave two new algorithms for incremental topological ordering. Their algorithm for the sparse case requires $\mathcal{O}(m^{3/2})$ time while their algorithm for the dense case requires $\mathcal{O}(n^{5/2})$ time. Bender et al. [7] gave an $\mathcal{O}(n^2 \log n)$ algorithm for this problem, thereby improving the results of Haeupler et al. for the dense case.

Acknowledgements

The authors are grateful to Telikepalli Kavitha, Irit Katriel, and Ulrich Meyer for various helpful discussions.

References

[1] D. Ajwani, T. Friedrich, Average-case analysis of online topological ordering, in: Proceedings of the International Symposium on Algorithms and Computation, (ISAAC '07), in: Lecture Notes in Computer Science, vol. 4835, Springer, 2007, pp. 464–475.
[2] D. Ajwani, T. Friedrich, U. Meyer, An $O(n^{2.75})$ algorithm for online topological ordering, in: Proceedings of the Scandinavian Workshop on Algorithm Theory, (SWAT '06), in: Lecture Notes in Computer Science, vol. 4059, 2006, pp. 53–64.
[3] D. Ajwani, T. Friedrich, U. Meyer, An $O(n^{2.75})$ algorithm for incremental topological ordering, ACM Trans. Algorithms 4 (2008) 1–14. Preliminary version appeared as [2].
[4] B. Alpern, R. Hoover, B.K. Rosen, P.F. Sweeney, F.K. Zadeck, Incremental evaluation of computational circuits, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, (SODA '90), 1990, pp. 32–42.
[5] G. Ausiello, G.F. Italiano, A. Marchetti-Spaccamela, U. Nanni, Incremental algorithms for minimal length paths, J. Algorithms 12 (1991) 615–638.
[6] A.B. Barak, P. Erdős, On the maximal number of strongly independent vertices in a random acyclic directed graph, SIAM Journal on Algeb. Discrete Methods 5 (1984) 508–514.
[7] M.A. Bender, J.T. Fineman, S. Gilbert, A new approach to incremental topological ordering, in: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '09), 2009, pp. 1108–1115.
[8] B. Bollobás, Degree sequences of random graphs, Discrete Math. 33 (1981) 1–19.
[9] B. Bollobás, Random Graphs, Cambridge Univ. Press, 2001.
[10] S. Cicerone, D. Frigioni, U. Nanni, F. Pugliese, A uniform approach to semi-dynamic problems on digraphs, Theor. Comput. Sci. 203 (1998) 69–90.
[11] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, The MIT Press, Cambridge, MA, 1989.
[12] P.F. Dietz, D.D. Sleator, Two algorithms for maintaining order in a list, in: Proceedings of the ACM Symposium on Theory of Computing (STOC '87), 1987, pp. 365–372.
[13] D. Eppstein, Z. Galil, G.F. Italiano, Dynamic graph algorithms, in: M.J. Atallah (Ed.), Algorithms and Theory of Computation Handbook, CRC Press, 1999, chapter 8.
[14] P. Erdős, A. Rényi, On random graphs, Publ. Math. Debrecen 6 (1959) 290–297.
[15] P. Erdős, A. Rényi, On the evolution of random graphs, Magyar Tud. Akad. Mat. Kutato Int. Kozl. 5 (1960) 17–61.
[16] D. Frigioni, A. Marchetti-Spaccamela, U. Nanni, Fully dynamic shortest paths and negative cycles detection on digraphs with arbitrary arc weights, in: Proceedings of the European Symposium on Algorithms, (ESA '98), in: Lecture Notes in Computer Science, vol. 1461, 1998, pp. 320–331.
[17] B. Haeupler, T. Kavitha, R. Mathew, S. Sen, R.E. Tarjan, Faster algorithms for incremental topological ordering, in: Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I (ICALP '08), 2008, pp. 421–433.
[18] I. Katriel, H.L. Bodlaender, Online topological ordering, ACM Trans. Algorithms 2 (2006) 364–379. Preliminary version appeared as [19].
[19] I. Katriel, H.L. Bodlaender, Online topological ordering, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA '05), 2005, pp. 443–450.
[20] A. Marchetti-Spaccamela, U. Nanni, H. Rohnert, On-line graph algorithms for incremental compilation, in: Proceedings of the Workshop on Graph-Theoretic Concepts in Computer Science, (WG '93), in: Lecture Notes in Computer Science, vol. 790, 1993, pp. 70–86.
[21] A. Marchetti-Spaccamela, U. Nanni, H. Rohnert, Maintaining a topological order under edge insertions, Inf. Process. Lett. 59 (1996) 53–58.
[22] S.M. Omohundro, C.-C. Lim, J. Bilmes, The sather language compiler/debugger implementation, Technical Report 92-017, Berkeley: International Computer Science Institute, 1992.
[23] D.J. Pearce, P.H.J. Kelly, A dynamic topological sort algorithm for directed acyclic graphs, J. Exp. Algorithmics 11 (2006) 1.7. Preliminary version appeared as [24].
[24] D.J. Pearce, P.H.J. Kelly, A dynamic algorithm for topologically sorting directed acyclic graphs, in: Proceedings of the Workshop on Experimental and Efficient Algorithms, (WEA '04), in: Lecture Notes in Computer Science, vol. 3059, 2004, pp. 383–398.
[25] D.J. Pearce, P.H.J. Kelly, C. Hankin, Online cycle detection and difference propagation: Applications to pointer analysis, Softw. Qual. J. 12 (2004) 311–337.
[26] B. Pittel, R. Tungol, A phase transition phenomenon in a random directed acyclic graph, Random Struct. Algorithms 18 (2001) 164–184.
[27] G. Ramalingam, T.W. Reps, On the computational complexity of dynamic graph problems, Theor. Comput. Sci. 158 (1996) 233–277.
[28] L. Roditty, U. Zwick, A fully dynamic reachability algorithm for directed graphs with an almost linear update timem in: Proceedings of the ACM Symposium on Theory of Computing (STOC '04), 2004, pp. 184–191.
[29] L. Roditty, U. Zwick, On dynamic shortest paths problems, in: Proceedings of the European Symposium on Algorithms, (ESA '04), in: Lecture Notes in Computer Science, vol. 3221, Springer, 2004, pp. 580–591.